NEW OR CHANGED CAPABILITIES

Summary

The following summarizes the capablities implemented in SIV:

NETWORK COMMUNICATIONS
(SRD, Section 1.9.1)

Provides Ethernet Connection to subsystem under test via the SPC LAN or equivalent LAN.

Supports Standard DSCC local facility protocol (890-131).

DATA BLOCK GENERATION
(SRD, Section 1.9.2)

Creates data blocks according to the formats and values described in the Rapid Interface Definition File (RID). RIDs reflect information taken from DSN Detailed Subsystem Interface Agreement documents (820-13 and 820-16).

Provides a means to generate interface data blocks such that data in each field assumes values that span the valid range specified in the interface agreement.

Provides a means to generate data that is outside the valid range specified in the interface agreement for each field.

Automates the generation of interface data blocks.

Sends interface data blocks to the subsystem under test.

Rates of sending interface data blocks shall be within the rate ranges specified in the interface agreement.
Not exactly. The metering of data is restricted to time delays between blocks with the range of $[0.1, 10^6)$ seconds.

Generates interface data blocks for multiple interfaces of one subsystem concurrently.

Generates raw data blocks from an ASCII data file. Allows for dynamic selection of data file to use.

DATA RECEPTION & VALIDATION
(SRD, Section 1.9.3)

Receives interface data blocks of 890-131 format as sent from the subsystem under test.

Performs validation on received data blocks as necessary to decode data block to the formats and values described in the interface agreement RID.
This is not implemented fully and is missing 2 data types (BCD & Fixed Point) and is poorly implemented for one other type (Signed).

Receives interface data blocks for multiple interfaces of one subsystem concurrently.

Prints validation results in readable ASCII report format (printable disk file).

Prints report of testing performed (autocontrol test script run log).

MONITOR AND CONTROL
(SRD, Section 1.9.4)

Controls the subsystem under test: i.e., monitor/control by DMC shall not be required for testing. Specifically, provides for FAT transfer, CCN transfer, OD forwarding, EN reception, and OD response reception.

Allows user to control testing or select displays via directives (according to 890-132 standard).

Creates a readable dump of blocks input or output to/from the subsystem under test (as requested by MID).
If by readable, you include the output of od(1) on the blocks, then yes this is true.

Provides means to control testing: i.e., start, stop, configure, change level of

reporting.

Captures responses from subsystem under test (event notices) and integrates them in SIV's event message log (tagged by subsystem DDC).

Provides means to modify interface data dynamically during testing.

Provides status information regarding SIV configuration, data generation, and data reception. Data flow statistics include block size, count, rate, and direction.

Provides a facility to repeat test steps under automatic control (via autocontrol test scripts).

Provides user with ability to specify configuration, RID, and test files to operate from.

Provides for dynamic visibility into data block contents.
If by visibility, you include the output of od(1) on the blocks, then yes this is true (dump).

OTHER MISCELLANEOUS
(SRD, Section 1.9.5)

SIV is ported to the following host platforms: ~~MODCOMP 9735 under REAL/IX;~~ SUN Sparc 10 Workstations under Solaris 2.3.

In addition to the above, SIV Build 1.2.5~~1.3.0~~ incorporates the following:

890-201 COMMUNICATIONS PROTOCOL.

Five new "keywords" were added to the interface definition file (*.rid).

KEYWORD	LEGAL VALUES
comm_protocol:	131 or 201

Identifies when 201 encapsulation is required.

data_begin:	none

Identifies the beginning of the data portion of 201 encapsulated messages (i.e., excludes 201 and/or OPS-6-8 header)

data_end:	none

Identifies the ending of the data portion of 201 encapsulated data (i.e., excludes the trailer and checksum).

data_type:	none

Identifies the 890-201 Standard Data Block data type.

sub_id:	890-132 Monitor Segment Number or 890-201 "data_type" 41 NOCC realtime packet identifier. Range 1 to 127.

When the "comm_protocol" is 131 and message_id is 19 (i.e., Monitor Data) this keyword designates the monitor segment number. This allows the SIV to extract multiple Monitor Data Segments from a single message.

When the "comm_protocol" is 201 and the "data_type" is 41 (NOCC realtime monitor data) this keyword designates the packet identifier.

New operator directive: D201 <GEN | RCV>
controls 201 stream routing tables. The SIV does not support simultaneous 890-201 generation and receiption.

where:

GEN	-	enable generation of 890-201 messages to the target
RCV	-	enable reception of 890-201 messages from the target machine.

DATA LOGGING

Provides the capability to record all data to a disk file for subsequent processing/validation.

A typical scenario would be to log the data for some period of time, terminate logging, then validate the all the recorded data. The SIV DOES NOT support the ability to simulataneously record and validate the same data stream. BecauseHowever, when "validation" reaches the "end-of-file" it automatically terminates validation.

New operator directive: LOG <stream> <E|D> [HDR]
where:

<stream>	-	identifies the stream to record. The stream name must correspond to a interface definition file (*.rid) from the CNF display.
E | D	-	E = enable logging D = disable logging
HDR	-	include the header and trailer in the log file. The "header/trailer" is defined as those fields prior to the keyword "data_begin" and after the keyword "data_end".

DATA VALIDATION

generates validation report. Performs range check against legal values. The range of legal values are defined in the interface definition file (*.rid). Data values that are out-of-range are flagged by "** ERR **" in the report.

operator directive syntax: VAL <stream> <E|D>

where:

<stream>	-	identifies the stream to validate. The stream name must correspond to a interface definition log file (*.log).
E | D	-	E = enable logging D = disable logging

NOTES:

1. The operator directive syntax remains the same; however, the data must be "logged" before it can be validated.

2. Validation processes the "*.log" file until an end-of-file is reached. Validation is automatically terminated upon EOF.

3. The validation report (*.val) is appended to each time validation is enabled. Also, the entire "*.log" file is validated from beginning to EOF upon invocation of VAL.

SEND OPERATOR DIRECTIVES TO THE TARGET MACHINE

generates an 890-132 operator directive message (mid=12) from the SIV to the target machine.

new operator directive: TGT <od and parameters>

NOTES:

1. requires at least one interface definition file (*.rid) that defines the LMC (process code 0x0) as either the "src_pc" or "dst_pc". If no "*.rid" exists with the LMC process code, this directive will "time-out" since there will not be a FAT entry, unless there exists a "*.rid" with the CMC process code (0xC10). In this case the directive will be sent with the CMC process code as the source. HINT: check the LDPST display to verify that the logical data path and FAT entry are valid.

GENERATE SUBSYSTEM DISPLAYS WITH GRAPHICS AT THE SIV

provides for graphical displays in an X-window environment.

New operator directives: XPSI <XT1 | XT2 | XT3 | XT4>

where:

XT1	-	file associated with "TERM SCREEN 1 ..." directive.
XT2	-	file associated with "TERM SCREEN 2 ..." directive.
XT3	-	file associated with "TERM SCREEN 3 ..." directive.
XT4	-	file associated with "TERM SCREEN 4 ..." directive.

New directive: TERM SCREEN <screen> PSI <XT1|XT2|XT3|XT4>

XT1	-	file associated with "XPSI XT1" directive.
XT2	-	file associated with "XPSI XT2" directive.
XT3	-	file associated with "XPSI XT3" directive.
XT4	-	file associated with "XPSI XT4" directive.

CONFIGURE THE SIV IN A LINK

provides the ability to use link process codes other than 1.

New operator directive parameter: CNF <subsystem> [LINK=<link_number>]

where:

subsystem	-	subsystem identifier for locating "cnfdir/<subsystem> .cnf" file and "riddir/<subsystem>/*.rid" files
<link_number>	-	range 1 to 8. Identifies link number to be used for generating the FAT and logical process codes.

Files Changed

None. (This is the initial version of the SIV software)

In addition to the above, SIV Build 1.3.0 incorporates the following:

The SIV build tree was redesigned so that building SIV does not require intimate knowledge of SIV. The versions are a good (and not the only) example. Prior to 1.3.0 the CM and developer needed to know which files contain the versions ids and how they must be edited. Now the only top-level file Version.h is updated and the "right" things happen. The makefiles handle all current build needs including dependency, lint, binary, and CM targets. The new-top level makefiles are

Makefile - contains all the actual targets and handles the recursive building.
Makefile.c - contains all the generic targets shared by many of the makefiles.
Makefile.h - contains the common macro definitions (except version ids) for all the other makefiles.
Version.h - contains the version id macro definitions used everywhere else.

The new secondary-level makefiles, */Makefile, were written to take advantage of the new top-level design.

PLEASE NOTE: all the high-level targets make use of symbolic links *except* for the CM targets. The CM targets (install, installhome, and snapshot) make copies of the files and are not sensitive to any changes in the SIV development, or MSW or 201 directories. The other high-level targets (root and work) *are* sensitive to any changes because of the symbolic links.

The SIV operational directory was redesigned, so that all the system files are installed in a standard third party software location and multiple user working directories are installed away from the system files with minimal diskspace usage and the level of file sharing/exclusion being defined by each user.

To support the op directory redesign, two new files were added to the top-level of the SIV system directory. ReadmeSiv.txt contains installation and some usage instructions. This file is also included in section 7.1.1.1 of the SIV SOM. Sivuser.mak is a makefile for creating the user working directories. It's usage is described in the ReadmeSiv.txt file.

Improved operability by replacing the two scripts "go.sh" and "cleanup.sh" with "siv.sh". Cleanup is completely automatic regardless if the user exits by signal (<ctrl-c>) or by the command "term abort". This script also handles metric data gathering by saving usage info to a file and if connected to the outside world, sending email to veregge@isds-server.jpl.nasa.gov.

The new support files used by "siv.sh" are "msw.sh", "cleanup_msw.sh", and "ftok". These support files are generic and replace the MSW scripts "go.sh" and "cleanup.sh", "rmshm.sh", and "mskill". The script "msw.sh" calls "cleanup_msw.sh" on termination, so the user need only be familiar with "msw.sh". The additional advantages to theses scripts are that only the IPCs created by this instance of running msw.sh are removed and only the processes created by this instance of running msw.sh are terminated with TERM signals (and it works every time, leaving no orphan processes or IPCs). These scripts also have more complete error checking, signal handling, and recovery than the scripts they replace. The sudo(1) usage has been reduced to one call to improve security and simplify turning the usage of sudo off. When running as root, new files owned by root will have their ownership changed to the user on termination.

New standalone tools include:

- ridlint - a RID file syntax-checker.
- xlate - an 890-16 interface definiton to RID file translator.
- ridedit.sh - a rid file editor/syntax-checker for beginning users.
- arp.sh - a utility that returns the ethernet addresses used by SIV in the format required by SIV.

The RID filre read code was completely rewritten, fixing all known bugs and adding error checking. SIV will no longer accept RID files with errors. For a complete list of the changes to the RID definition see Section 6.4 - *Changes to the RID Definition* in the SIV SOM in S:\CWO_12/docs/ug/sec6.doc. This eliminated the use of the file lib/sivrd.c and added the new files datatype.c, datatype.h, library.h, pr_block_def.c, pr_msg.c, pr_stream.c, rid_file.c, rid_file.h, rid_fread.c, str_conv.c, str_conv.h, str_par.c, and str_par.h (in the directories lib or include).

The validation half of dataval was rewritten fixing all known bugs, adding the missing support for using the numeric format specified by the RID files, and adding the missing support for the BCD and Fixed Point data types (all in the function Dataval(), in the file dataval/dataval.c). *However, the data conversion half of dataval is still missing the support for those same two types and only partially supports the signed data type.*

The RID file reading portion of sivmgr was rewritten to use the new RID file reading code in the library (in the function getridinfo() in the file sivmgr/mgrfat.c). No algorithmic changes were made other than those already contained in the RID file library code.

Directory	Filename	Status	Reason
(main)	build.dir	deleted	
build	*riv.mak	deleted	replaced with Makefile, Makefile.h, Makefile.c, Version.h in (main)
build	rivenv.inc	deleted	
csen	csen.mak	deletedrenamed	replaced withto Mmakefile.
data	cnfdir.dir, riddir.dir, tstdir.dir, dspdir.dir	renamed	to cnf.dir, rid.dir, tst.dir and dsp.dir
data	*.raw, *.txt, *.sn, *.siv, *.rx, *.msw, *.dat	moved	to subdirectories data.dat, data.raw, data.old or data.txt
data	dbgdir.dir	deleted	contents moved to subdirectory data/dat
dbgdir	ridctl.ddx	moveddeleted	to subdirectory data/dat
data	shmdir.dir	deleted	contents moved to subdirectory data/dat
data.shmdir	riv.shm	moveddeleted	to subdirectory data/dat
data.riddir	gen201.dir	renamed	to 201.dir
data.riddir.gen201	dfl11b.rid	moved	to data.rid.201
data.riddir.gen201	dfl201e8.rid	moved	to data.rid.201
data.riddir.gen201	dfl201p.rid	moved	to data.rid.201
data.riddir.mda	bvrmda[61,62,63,.rid, ccnmda.rid, etcmda[71,72,73].rid, mdabvr40.rid, mdalmc[01,05].rid, rcvmda30.rid, txrmda[09,10].rid	moved	to data.rid.mda0 and/or data.rid.mda
data.rid.mda	ccnmda[1,2].rid, etcseg[71,72,73, 75].rid, rcvseg30.rid, txrseg[10,9].rid	added	not to be confused with data.riddir.mda from the previous SIV version
data.riddir.test	tbfssy4.rid	deleted	duplicate of file data/rid/t5/ tbfssy4.rid
datagen	datagen.mak	replacedrenamed	withto Mmakefile.
datarcv	datarcv.mak	replacedrenamed	withto Mmakefile.
dataval	dataval.mak	replacedrenamed	withto Mmakefile.
isb_dmp	isb_dmp.mak	replacedrenamed	withto Mmakefile.
rawgen	rawgen.mak	replacedrenamed	withto Mmakefile.
(main)	rivlib.dir	renamed	to lib.dir
rivlib	rivlib.mak	replacedrenamed	withto Mmakefile.
rivlib	rivset.c	deleted	
rivshm	rivshm.mak, rivshmrx.mak, rivshmsn.mak	replaced	with Mmakefile.
sivmgr	sivmgr.mak	replacedrenamed	withto Mmakefile.
util	util.mak	replacedrenamed	withto Mmakefile.
(main)	ftok.dir	added	
(main)	.rid	added	
(main)	ridlint.dir	added	
(main)	tmp.dir	added	
(main)	Vversion.h	added	see build
(main)	Makefile.h	added	see build
(main)	Makefile.c	added	see build
(main)	Makefile	added	see build
(main)	xlate.dir	added	
(main)	tools.dir	added	

Directory	Filename	Status	Reason
data	raw.dir	added	(6 files moved from build125 ‘data’)
data	dat.dir	added	(7 of 10 files moved from build125 ‘data’)
data	top.dir	added	
data	old.dir	added	(5 files moved from build125 ‘data’)
data.dat	sysinit.ini	added	
data.dat	ridctl.ddx	added	
data.dat	riv.shm	added	
data.rid	201.dir	added	
data.rid	readme.	added	
data.rid	mda0.dir	added	
data.top	Rreadmesiv.txt	added	
data.top	Sivuser.mak	added	
~~data.tst~~	~~atest.log~~	~~added~~	
ftok	ftok.c	added	
ftok	Mmakefile.	added	
include	datatype.h	added	
include	library.h	added	
include	rid_file.h	added	
include	str_conv.h	added	
include	str_par.h	added	
include	–datatype.h	added	
lib	datatype.c	added	
lib	pr_block_def.c	added	
lib	str_par.c	added	
lib	pr_msg.c	added	
lib	pr_stream.c	added	
lib	rid_file.c	added	
lib	rid_fread.c	added	
lib	str_conv.c	added	
ridlint	(all)	added	(19 files)
ridlint.bak	(all)	added	(3 files)
ridlint.debug3	(all)	added	(15 files)
ridlint.debug3.201	(all)	added	(3 files)
ridlint.debug3.acg	(all)	added	(15 files)
ridlint.debug3.apa	(all)	added	(10 files)
ridlint.debug3.bvr	(all)	added	(45 files)
ridlint.debug3.cmc	(all)	added	(6 files)
ridlint.debug3.dgt	(all)	added	(20 files)
ridlint.debug3.mda	(all)	added	(12 files)
ridlint.debug3.mda0	(all)	added	(13 files)
ridlint.debug3.mpa	(all)	added	(5 files)
ridlint.debug3.ots	(all)	added	(42 files)
ridlint.debug3.rsc	(all)	added	(6 files)
ridlint.debug3.scp	(all)	added	(3 files)
ridlint.debug3.t5	(all)	added	(15 files)
ridlint.debug3.test	(all)	added	(14 files)
ridlint.debug3.ugc	(all)	added	(15 files)
ridlint.rid	(all)	added	(16 files)
ridlint.rid.201	(all)	added	(3 files)

Directory	Filename	Status	Reason
ridlint.rid.acg	(all)	added	(14 files)
ridlint.rid.apa	(all)	added	(10 files)
ridlint.rid.bvr	(all)	added	(45 files)
ridlint.rid.cmc	(all)	added	(6 files)
ridlint.rid.dgt	(all)	added	(19 files)
ridlint.rid.mda	(all)	added	(12 files)
ridlint.rid.mda0	(all)	added	(13 files)
ridlint.rid.mpa	(all)	added	(3 files)
ridlint.rid.ots	(all)	added	(40 files)
ridlint.rid.rsc	(all)	added	(6 files)
ridlint.rid.scp	(all)	added	(2 files)
ridlint.rid.t5	(all)	added	(11 files)
ridlint.rid.test	(all)	added	(12 files)
ridlint.rid.ugc	(all)	added	(15 files)
tmp	etcseg75.log	added	
tmp	etcseg75.ts	added	
tmp	etcseg75.oldlog	added	
tmp	etcseg75.val	added	
tmp	etcseg75.oldts	added	
tmp	etcseg75.oldval	added	
tools	arp.sh	added	
tools	cleanup_msw.sh	added	
tools	Mmakefilie.	added	
tools	msw.sh	added	
tools	ridedit.sh	added	
tools	siv.sh	added	
xlate	.rid	added	
xlate	Mmakefile.	added	
xlate	xlate.	added	
xlate	tenth.	added	
xlate	test.dir	added	
xlate.test	10test.	added	
xlate.test	all_types.id	added	
xlate.test	ascii_flt_file.id	added	
xlate.test	binary_pt_file.id	added	
xlate.test	bool_file.id	added	
xlate.test	break.id	added	
xlate.test	char_file.id	added	
xlate.test	chksum_file.id	added	
xlate.test	errors.id	added	
xlate.test	float_file.id	added	
xlate.test	if.id	added	
xlate.test	int_file.id	added	
xlate.test	sub.id	added	
xlate.test	substitute.id	added	
xlate.test	substitute2.id	added	
xlate.test	time_file.id	added	
xlate.test	tr2.id	added	
xlate.test	trk_time.id	added	